Proceedings of the 40th Annual Meeting of the Association for
Computational Linguistics (ACL), Philadelphia, July 2002, pp. 400-408.

Translating Named Entities Using Monolingual and Bilingual Resources

Yaser Al-Onaizan and **Kevin Knight**
Information Sciences Institute
University of Southern California
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292
{yaser,knight}@isi.edu

Abstract

Named entity phrases are some of the most difficult phrases to translate because new phrases can appear from nowhere, and because many are domain specific, not to be found in bilingual dictionaries. We present a novel algorithm for translating named entity phrases using easily obtainable monolingual and bilingual resources. We report on the application and evaluation of this algorithm in translating Arabic named entities to English. We also compare our results with the results obtained from human translations and a commercial system for the same task.

1 Introduction

Named entity phrases are being introduced in news stories on a daily basis in the form of personal names, organizations, locations, temporal phrases, and monetary expressions. While the identification of named entities in text has received significant attention (e.g., Mikheev et al. (1999) and Bikel et al. (1999)), translation of named entities has not. This translation problem is especially challenging because new phrases can appear from nowhere, and because many named-entities are domain specific, not to be found in bilingual dictionaries.

A system that specializes in translating named entities such as the one we describe here would be an important tool for many NLP applications. Statistical machine translation systems can use such a system as a component to handle phrase translation in order to improve overall translation quality. Cross-Lingual Information Retrieval (CLIR) systems could identify relevant documents based on translations of named entity phrases provided by such a system. Question Answering (QA) systems could benefit substantially from such a tool since the answer to many factoid questions involve named entities (e.g., answers to *who* questions usually involve **Persons**/**Organizations**, *where* questions involve **Locations**, and *when* questions involve **Temporal Expressions**).

In this paper, we describe a system for Arabic-English named entity translation, though the technique is applicable to any language pair and does not require especially difficult-to-obtain resources.

The rest of this paper is organized as follows. In Section 2, we give an overview of our approach. In Section 3, we describe how translation candidates are generated. In Section 4, we show how monolingual clues are used to help re-rank the translation candidates list. In Section 5, we describe how the candidates list can be extended using contextual information. We conclude this paper with the evaluation results of our translation algorithm on a test set. We also compare our system with human translators and a commercial system.

2 Our Approach

The frequency of named-entity phrases in news text reflects the significance of the events they are associated with. When translating named entities in news stories of international importance, the same event will most likely be reported in many languages including the target language. Instead of having to come up with translations for the named entities often with many unknown words in one document, sometimes it is easier for a human to find a document in the target language that is similar to, but not necessarily a translation of, the original document and then extract the translations. Let's illustrate this idea with the following example:

2.1 Example

We would like to translate the named entities that appear in the following Arabic excerpt:

وستجرى عشر عمليات عام ٢٠٠١م من ابريل
(نيسان) وحتى نوفمبر (تشرين الثاني) حول خزان
تشوزين وفي منطقتي البحث الحاليتين في مقاطعتي
اونسان وكوجانج الواقعتين على بعد ٩٦ كيلومترا
تقريبا شمالي بيونغ يانغ.

The Arabic newspaper article from which we extracted this excerpt is about negotiations between the US and North Korean authorities regarding the search for the remains of US soldiers who died during the Korean war.

We presented the Arabic document to a bilingual speaker and asked them to translate the locations "تشوزين *tšwzyn* خزان *ḫzān*", "اونسان *āwnsān*", and "كوجانج *kwǧānǧ*." The translations they provided were *Chozin Reserve*, *Onsan*, and *Kojanj*. It is obvious that the human attempted to sound out names and despite coming close, they failed to get them correctly as we will see later.

When translating unknown or unfamiliar names, one effective approach is to search for an English document that discusses the same subject and then extract the translations. For this example, we start by creating the following Web query that we use with the search engine:

Search Query 1: *soldiers remains*, *search*, *North Korea*, and *US*.

This query returned many hits. The top document returned by the search engine[1] we used contained the following paragraph:

> The targeted area is near **Unsan**, which saw several battles between the U.S. Army's 8th Cavalry regiment and Chinese troops who launched a surprise offensive in late 1950.

This allowed us to create a more precise query by adding *Unsan* to the search terms:

Search Query 2: *soldiers remains*, *search*, *North Korea*, *US*, and *Unsan*.

This search query returned only 3 documents. The first one is the above document. The third is the top level page for the second document. The second document contained the following excerpt:

> Operations in 2001 will include areas of investigation near Kaechon, approximately 18 miles south of **Unsan** and **Kujang**. Kaechon includes an area nicknamed the "Gauntlet," where the U.S. Army's 2nd Infantry Division conducted its famous fighting withdrawal along a narrow road through six miles of Chinese ambush positions during November and December 1950. More than 950 missing in action soldiers are believed to be located in these three areas.
>
> The **Chosin Reservoir** campaign left approximately 750 Marines and soldiers missing in action from both the east and west sides of the reservoir in northeastern North Korea.

This human translation method gives us the correct translation for the names we are interested in.

2.2 Two-Step Approach

Inspired by this, our goal is to tackle the named entity translation problem using the same approach described above, but fully automatically and using the least amount of hard-to-obtain bilingual resources.

As shown in Figure 1, the translation process in our system is carried out in two main steps. Given a named entity in the source language, our translation algorithm first generates a ranked list of translation candidates using bilingual and monolingual resources, which we describe in the Section 3. Then, the list of candidates is re-scored using different monolingual clues (Section 4).

[1] http://www.google.com/

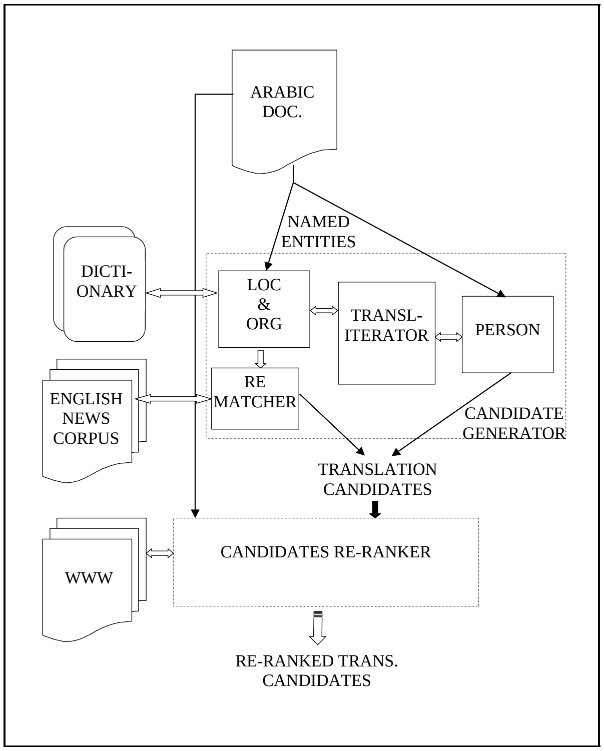

Figure 1: A sketch of our named entity translation system.

3 Producing Translation Candidates

Named entity phrases can be identified fairly accurately (e.g., Bikel et al. (1999) report an F-MEASURE of 94.9%). In addition to identifying phrase boundaries, named-entity identifiers also provide the category and sub-category of a phrase (e.g., **ENTITY NAME**, and **PERSON**). Different types of named entities are translated differently and hence our candidate generator has a specialized module for each type. Numerical and temporal expressions typically use a limited set of vocabulary words (e.g., names of months, days of the week, etc.) and can be translated fairly easily using simple translation patterns. Therefore, we will not address them in this paper. Instead we will focus on person names, locations, and organizations. But before we present further details, we will discuss how words can be transliterated (i.e., "sounded-out"), which is a crucial component of our named entity translation algorithm.

3.1 Transliteration

Transliteration is the process of replacing words in the source language with their approximate phonetic or spelling equivalents in the target language. Transliteration between languages that use similar alphabets and sound systems is very simple. However, transliterating names from Arabic into English is a non-trivial task, mainly due to the differences in their sound and writing systems. Vowels in Arabic come in two varieties: long vowels and short vowels. Short vowels are rarely written in Arabic in newspaper text, which makes pronunciation and meaning highly ambiguous. Also, there is no one-to-one correspondence between Arabic sounds and English sounds. For example, English **P** and **B** are both mapped into Arabic "ب *b*"; Arabic "ح *ḥ*" and "هـ *h*-" into English **H**; and so on.

Stalls and Knight (1998) present an Arabic-to-English back-transliteration system based on the source-channel framework. The transliteration process is based on a generative model of how an English name is transliterated into Arabic. It consists of several steps, each is defined as a probabilistic model represented as a finite state machine. First, an English word is generated according to its unigram probabilities $P(w)$. Then, the English word is pronounced with probability $P(e|w)$, which is collected directly from an English pronunciation dictionary. Finally, the English phoneme sequence is converted into Arabic writing with probability $P(a|e)$. According to this model, the transliteration probability is given by the following equation:

$$P_p(w|a) \simeq \sum_{\forall e} P(w)P(e|w)P(a|e) \qquad (1)$$

The transliterations proposed by this model are generally accurate. However, one serious limitation of this method is that only English words with known pronunciations can be produced. Also, human translators often transliterate words based on how they are spelled in the source language. For example, *Graham* is transliterated into Arabic as "غراهام *ġrāhām*" and not as "غرام *ġrām*". To address these limitations, we extend this approach by using a new spelling-based model in addition to the phonetic-based model.

The spelling-based model we propose (described in detail in (Al-Onaizan and Knight, 2002)) directly maps English letter sequences into Arabic letter sequences with probability $P(a|w)$, which are trained on a small English/Arabic name list without the need for English pronunciations. Since no pronunciations are needed, this list is easily obtainable for many language pairs. We also extend the model $P(w)$ to include a letter trigram model in addition to the word unigram model. This makes it possible to generate words that are not already defined in the word unigram model. The transliteration score according to this model is given by:

$$P_s(w|a) \simeq P(w)P(a|w) \quad (2)$$

The phonetic-based and spelling-based models are combined into a single transliteration model. The transliteration score for an English word w given an Arabic word a is a linear combination of the phonetic-based and the spelling-based transliteration scores as follows:

$$P(w|a) = \lambda P_s(w|a) + (1 - \lambda)P_p(w|a) \quad (3)$$

3.2 Producing Candidates for Person Names

Person names are almost always transliterated. The translation candidates for typical person names are generated using the transliteration module described above. Finite-state devices produce a lattice containing all possible transliterations for a given name. The candidate list is created by extracting the n-best transliterations for a given name. The score of each candidate in the list is the transliteration probability as given by Equation 3. For example, the name "كلينتون *klyntwn* بيل *byl*" is transliterated into: *Bell Clinton*, *Bill Clinton*, *Bill Klington*, etc.

3.3 Producing Candidates for Location and Organization Names

Words in organization and location names, on the other hand, are either translated (e.g., "خزان *ḫzān*" as *Reservoir*) or transliterated (e.g., "تشوزين *tšwzyn*" as *Chosin*), and it is not clear when a word must be translated and when it must be transliterated. So to generate translation candidates for a given phrase f, words in the phrase are first translated using a bilingual dictionary and they are also transliterated. Our candidate generator combines the dictionary entries and n-best transliterations for each word in the given phrase into a regular expression that accepts all possible permutations of word translation/transliteration combinations. In addition to the word transliterations and translations, English zero-fertility words (i.e., words that might not have Arabic equivalents in the named entity phrase such as *of* and *the*) are considered. This regular expression is then matched against a large English news corpus. All matches are then scored according to their individual word translation/transliteration scores. The score for a given candidate e is given by a modified IBM Model 1 probability (Brown et al., 1993) as follows:

$$\begin{aligned} P(e|f) &= \alpha \sum_{\forall a} P(e, a|f) & (4) \\ &= \alpha \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(f_j|e_{a_j}) & (5) \end{aligned}$$

where l is the length of e, m is the length of f, α is a scaling factor based on the number of matches of e found, and a_j is the index of the English word aligned with f_j according to alignment a. The probability $t(e_{a_j}|f_j)$ is a linear combination of the transliteration and translation score, where the translation score is a uniform probability over all dictionary entries for f_j.

The scored matches form the list of translation candidates. For example, the candidate list for "الخنازير *al-ḫnāzyr* خليج *ḫlyǧ*" includes *Bay of Pigs* and *Gulf of Pigs*.

4 Re-Scoring Candidates

Once a ranked list of translation candidates is generated for a given phrase, several monolingual English resources are used to help re-rank the list. The candidates are re-ranked according to the following equation:

$$S_{new}(c) = S_{old}(c) \times RF(c) \quad (6)$$

where $RF(c)$ is the re-scoring factor used.

Straight Web Counts: (Grefenstette, 1999) used phrase Web frequency to disambiguate possible English translations for German and Spanish compound nouns. We use normalized Web counts of named entity phrases as the first re-scoring factor used to rescore translation candidates. For the "كلينتون *klyntwn* بيل *byl*" example, the top two translation candidates are *Bell Clinton* with transliteration score 1.1×10^{-09} and *Bill Clinton* with score 6.7×10^{-10}. The Web frequency counts of these two names are: 146 and $840, 844$ respectively. This gives us revised scores of 1.9×10^{-13} and 6.68×10^{-10}, respectively, which leads to the correct translation being ranked highest.

It is important to consider counts for the full name rather than the individual words in the name to get accurate counts. To illustrate this point consider the person name “كيل *kyl* جون *ǧwn*.” The transliteration module proposes *Jon* and *John* as possible transliterations for the first name, and *Keele* and *Kyl* among others for the last name. The normalized counts for the individual words are: (*John*, 0.9269), (*Jon*, 0.0688), (*Keele*, 0.0032), and (*Kyl*, 0.0011). To use these normalized counts to score and rank the first name/last name combinations in a way similar to a unigram language model, we would get the following name/score pairs: (*John Keele*, 0.003), (*John Kyl*, 0.001), (*Jon Keele*, 0.0002), and (*Jon Kyl*, $7.5{\times}10^{-5}$). However, the normalized phrase counts for the possible full names are: (*Jon Kyl*, 0.8976), (*John Kyl*, 0.0936), (*John Keele*, 0.0087), and (*Jon Keele*, 0.0001), which is more desirable as *Jon Kyl* is an often-mentioned US Senator.

Co-reference: When a named entity is first mentioned in a news article, typically the full form of the phrase (e.g., the full name of a person) is used. Later references to the name often use a shortened version of the name (e.g, the last name of the person). Shortened versions are more ambiguous by nature than the full version of a phrase and hence more difficult to translate. Also, longer phrases tend to have more accurate Web counts than shorter ones as we have shown above. For example, the phrase “النواب *al-nwāb* مجلس *mǧls*” is translated as *the House of Representatives*. The word “المجلس *al-mǧls*”[2] might be used for later references to this phrase. In that case, we are confronted with the task of translating “المجلس *al-mǧls*” which is ambiguous and could refer to a number of things including: *the Council* when referring to “الأمن *al-ʾmn* مجلس *mǧls*” (*the Security Council*); *the House* when referring to ‘النواب *al-nwāb* مجلس *mǧls*” (*the House of Representatives*); and as *the Assembly* when referring to “الأمة *al-ʾmt* مجلس *mǧls*” (*National Assembly*).

If we are able to determine that in fact it was referring to *the House of Representatives*, then, we can translate it accurately as *the House*. This can be done by comparing the shortened phrase with the rest of the named entity phrases of the same type. If the shortened phrase is found to be a sub-phrase of only one other phrase, then, we conclude that the shortened phrase is another reference to the same named entity. In that case we use the counts of the longer phrase to re-rank the candidates of the shorter one.

Contextual Web Counts: In some cases straight Web counting does not help the re-scoring. For example, the top two translation candidates for “مارون *mārwn* دونالد *dwnāld*” are *Donald Martin* and *Donald Marron*. Their straight Web counts are 2992 and 2509, respectively. These counts do not change the ranking of the candidates list. We next seek a more accurate counting method by counting phrases only if they appear within a certain context. Using search engines, this can be done using the boolean operator **AND**. For the previous example, we use *Wall Street* as the contextual information In this case we get the counts 15 and 113 for *Donald Martin* and *Donald Marron*, respectively. This is enough to get the correct translation as the top candidate.

The challenge is to find the contextual information that provide the most accurate counts. We have experimented with several techniques to identify the contextual information automatically. Some of these techniques use document-wide contextual information such as the title of the document or select key terms mentioned in the document. One way to identify those key terms is to use the *tf.idf* measure. Others use contextual information that is local to the named entity in question such as the n words that precede and/or succeed the named entity or other named entities mentioned closely to the one in question.

5 Extending the Candidates List

The re-scoring methods described above assume that the correct translation is in the candidates list. When it is not in the list, the re-scoring will fail. To address this situation, we need to extrapolate from the candidate list. We do this by searching for the correct translation rather than generating it. We do that by using sub-phrases from the candidates list

[2] “المجلس *al-mǧls*” is the same word as “مجلس *mǧls*” but with the definite article ال *a-* attached.

or by searching for documents in the target language similar to the one being translated. For example, for a person name, instead of searching for the full name, we search for the first name and the last name separately. Then, we use the IdentiFinder named entity identifier (Bikel et al., 1999) to identify all named entities in the top n retrieved documents for each sub-phrase. All named entities of the type of the named entity in question (e.g., PERSON) found in the retrieved documents and that contain the sub-phrase used in the search are scored using our transliteration module and added to the list of translation candidates, and the re-scoring is repeated.

To illustrate this method, consider the name "عنان *ʿnān* كوفي *kwfy*." Our translation module proposes: *Coffee Annan*, *Coffee Engen*, *Coffee Anton*, *Coffee Anyone*, and *Covey Annan* but not the correct translation **Kofi Annan**. We would like to find the most common person names that have either one of *Coffee* or *Covey* as a first name; or *Annan*, *Engen*, *Anton*, or *Anyone* as a last name. One way to do this is to search using wild cards. Since we are not aware of any search engine that allows wild-card Web search, we can perform a wild-card search instead over our news corpus. The problem is that our news corpus is dated material, and it might not contain the information we are interested in. In this case, our news corpus, for example, might predate the appointment of **Kofi Annan** as the Secretary General of the UN. Alternatively, using a search engine, we retrieve the top n matching documents for each of the names *Coffee*, *Covey*, *Annan*, *Engen*, *Anton*, and *Anyone*. All person names found in the retrieved documents that contain any of the first or last names we used in the search are added to the list of translation candidates. We hope that the correct translation is among the names found in the retrieved documents. The rescoring procedure is applied once more on the expanded candidates list. In this example, we add *Kofi Annan* to the candidate list, and it is subsequently ranked at the top.

To address cases where neither the correct translation nor any of its sub-phrases can be found in the list of translation candidates, we attempt to search for, instead of generating, translation candidates. This can be done by searching for a document in the target language that is similar to the one being translated from the source language. This is especially useful when translating named entities in news stories of international importance where the same event will most likely be reported in many languages including the target language. We currently do this by repeating the extrapolation procedure described above but this time using contextual information such as the title of the original document to find similar documents in the target language. Ideally, one would use a Cross-Lingual IR system to find relevant documents more successfully.

6 Evaluation and Discussion

6.1 Test Set

This section presents our evaluation results on the named entity translation task. We compare the translation results obtained from human translations, a commercial MT system, and our named entity translation system. The evaluation corpus consists of two different test sets, a **development test set** and a **blind test set**. The first set consists of 21 Arabic newspaper articles taken from the political affairs section of the daily newspaper Al-Riyadh. Named entity phrases in these articles were hand-tagged according to the MUC (Chinchor, 1997) guidelines. They were then translated to English by a bilingual speaker (a native speaker of Arabic) given the text they appear in. The Arabic phrases were then paired with their English translations.

The blind test set consists of 20 Arabic newspaper articles that were selected from the political section of the Arabic daily Al-Hayat. The articles have already been translated into English by professional translators.[3] Named entity phrases in these articles were hand-tagged, extracted, and paired with their English translations to create the blind test set.

Table 1 shows the distribution of the named entity phrases into the three categories **PERSON**, **ORGANIZATION** , and **LOCATION** in the two data sets.

The English translations in the two data sets were reviewed thoroughly to correct any wrong translations made by the original translators. For example, to find the correct translation of a politician's name, official government web pages were used to find the

[3] The Arabic articles along with their English translations were part of the FBIS 2001 Multilingual corpus.

Test Set	PERSON	ORG	LOC
Development	33.57	25.62	40.81
Blind	28.38	21.96	49.66

Table 1: The distribution of named entities in the test sets into the categories **PERSON**, **ORGANIZATION** , and **LOCATION**. The numbers shown are the ratio of each category to the total.

correct spelling. In cases where the translation could not be verified, the original translation provided by the human translator was considered the “correct“ translation. The Arabic phrases and their correct translations constitute the gold-standard translation for the two test sets.

According to our evaluation criteria, only translations that match the gold-standard are considered as correct. In some cases, this criterion is too rigid, as it will consider perfectly acceptable translations as incorrect. However, since we use it mainly to compare our results with those obtained from the human translations and the commercial system, this criterion is sufficient. The actual accuracy figures might be slightly higher than what we report here.

6.2 Evaluation Results

In order to evaluate human performance at this task, we compared the translations by the original human translators with the correct translations on the gold-standard. The errors made by the original human translators turned out to be numerous, ranging from simple spelling errors (e.g., *Custa Rica* vs. *Costa Rica*) to more serious errors such as transliteration errors (e.g., *John Keele* vs. *Jon Kyl*) and other translation errors (e.g., *Union Reserve Council* vs. *Federal Reserve Board*).

The Arabic documents were also translated using a commercial Arabic-to-English translation system.[4] The translation of the named entity phrases are then manually extracted from the translated text. When compared with the gold-standard, nearly half of the phrases in the development test set and more than a third of the blind test were translated incorrectly by the commercial system. The errors can be classified into several categories including: **poor transliterations** (e.g., *Koln Baol* vs. *Colin Powell*), **translating a name instead of sounding it out** (e.g., *O'Neill's urine* vs. *Paul O'Neill*), **wrong translation** (e.g., *Joint Corners Organization* vs. *Joint Chiefs of Staff*) or **wrong word order** (e.g.,*the Church of the Orthodox Roman*).

Table 2 shows a detailed comparison of the translation accuracy between our system, the commercial system, and the human translators. The translations obtained by our system show significant improvement over the commercial system. In fact, in some cases it outperforms the human translator. When we consider the top-20 translations, our system's overall accuracy (84%) is higher than the human's (75.3%) on the blind test set. This means that there is a lot of room for improvement once we consider more effective re-scoring methods. Also, the top-20 list in itself is often useful in providing phrasal translation candidates for general purpose statistical machine translation systems or other NLP systems.

The strength of our translation system is in translating person names, which indicates the strength of our transliteration module. This might also be attributed to the low named entity coverage of our bilingual dictionary. In some cases, some words that need to be translated (as opposed to transliterated) are not found in our bilingual dictionary which may lead to incorrect location or organization translations but does not affect person names. The reason word translations are sometimes not found in the dictionary is not necessarily because of the spotty coverage of the dictionary but because of the way we access definitions in the dictionary. Only shallow morphological analysis (e.g., removing prefixes and suffixes) is done before accessing the dictionary, whereas a full morphological analysis is necessary, especially for morphologically rich languages such as Arabic. Another reason for doing poorly on organizations is that acronyms and abbreviations in the Arabic text (e.g., “واس *wās*,” *the Saudi Press Agency*) are currently not handled by our system.

The blind test set was selected from the FBIS 2001 Multilingual Corpus. The FBIS data is collected by the Foreign Broadcast Information Service for the benefit of the US government. We suspect that the human translators who translated the documents into English are somewhat familiar with the genre of the articles and hence the named entities

[4] We used Sakhr's Web-based translation system available at http://tarjim.ajeeb.com/.

System	**Accuracy (%)**			
	PERSON	**ORG**	**LOC**	**Overall**
Human	60.00	71.70	86.10	73.70
Sakhr	29.47	51.72	72.73	52.80
Top-1 Results	77.20	43.30	69.00	65.20
Top-20 Results	84.80	55.00	70.50	71.33

(a) Results on the **Development Test Set**

System	**Accuracy (%)**			
	PERSON	**ORG**	**LOC**	**Overall**
Human	67.89	42.20	94.68	75.30
Sakhr	47.71	36.05	80.80	61.30
Top-1 Results	64.24	51.00	86.68	72.57
Top-20 Results	78.84	70.80	92.86	84.00

(b) Results on the **Blind Test Set**

Table 2: A comparison of translation accuracy for the human translator, commercial system, and our system on the development and blind test sets. Only a match with the translation in the gold-standard is considered a correct translation. The **human translator** results are obtained by comparing the translations provided by the original human translator with the translations in the gold-standard. The **Sakhr** results are for the Web version of Sakhr's commercial system. The **Top-1** results of our system considers whether the correct answer is the top candidate or not, while the **Top-20** results considers whether the correct answer is among the top-20 candidates. **Overall** is a weighted average of the three named entity categories.

Module	**Accuracy (%)**			
	PERSON	**ORG**	**LOC**	**Overall**
Candidate Generator	59.85	31.67	54.00	49.96
Straight Web Counts	75.76	37.97	63.37	61.02
Contextual Web Counts	75.76	39.17	67.50	63.01
Co-reference	77.20	43.30	69.00	65.20

(a) Results on the **Development test set**

Module	**Accuracy (%)**			
	PERSON	**ORG**	**LOC**	**Overall**
Candidate Generator	54.33	51.55	85.75	69.44
Straight Web Counts	61.00	46.60	86.68	70.66
Contextual Web Counts	62.50	45.34	85.75	70.40
Co-reference	64.24	51.00	86.68	72.57

(b) Results on the **Blind Test Set**

Table 3: This table shows the accuracy after each translation module. The modules are applied incrementally. **Straight Web Counts** re-score candidates based on their Web counts. **Contextual Web Counts** uses Web counts within a given context (we used here title of the document as the contextual information). In **Co-reference**, if the phrase to be translated is part of a longer phrase then we use the the ranking of the candidates for the longer phrase to re-rank the candidates of the short one, otherwise we leave the list as is.

that appear in the text. On the other hand, the development test set was randomly selected by us from our pool of Arabic articles and then submitted to the human translator. Therefore, the human translations in the blind set are generally more accurate than the human translations in the development test. Another reason might be the fact that the human translator who translated the development test is not a professional translator.

The only exception to this trend is organizations. After reviewing the translations, we discovered that many of the organization translations provided by the human translator in the blind test set that were judged incorrect were acronyms or abbreviations for the full name of the organization (e.g., *the INC* instead of *the Iraqi National Congress*).

6.3 Effects of Re-Scoring

As we described earlier in this paper, our translation system first generates a list of translation candidates, then re-scores them using several re-scoring methods. The list of translation candidates we used for these experiments are of size 20. The re-scoring methods are applied incrementally where the reranked list of one module is the input to the next module. Table 3 shows the translation accuracy after each of the methods we evaluated.

The most effective re-scoring method was the simplest, the straight Web counts. This is because re-scoring methods are applied incrementally and straight Web counts was the first to be applied, and so it helps to resolve the "easy" cases, whereas the other methods are left with the more "difficult" cases. It would be interesting to see how rearranging the order in which the modules are applied might affect the overall accuracy of the system.

The re-scoring methods we used so far are in general most effective when applied to person name translation because corpus phrase counts are already being used by the candidate generator for producing candidates for locations and organizations, but not for persons. Also, the re-scoring methods we used were initially developed and applied to person names. More effective re-scoring methods are clearly needed especially for organization names. One method is to count phrases only if they are tagged by a named entity identifier with the same tag we are interested in. This way we can eliminate counting wrong translations such as *enthusiasm* when translating "حماس *ḥmās*" (*Hamas*).

7 Conclusion and Future Work

We have presented a named entity translation algorithm that performs at near human translation accuracy when translating Arabic named entities to English. The algorithm uses very limited amount of hard-to-obtain bilingual resources and should be easily adaptable to other languages. We would like to apply to other languages such as Chinese and Japanese and to investigate whether the current algorithm would perform as well or whether new algorithms might be needed.

Currently, our translation algorithm does not use any dictionary of named entities and they are translated on the fly. Translating a common name incorrectly has a significant effect on the translation accuracy. We would like to experiment with adding a small named entity translation dictionary for common names and see if this might improve the overall translation accuracy.

Acknowledgments

This work was supported by DARPA-ITO grant N66001-00-1-9814.

References

Yaser Al-Onaizan and Kevin Knight. 2002. Machine Transliteration of Names in Arabic Text. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*.

Daniel M. Bikel, Richard Schwartz, and Ralph M. Weischedel. 1999. An algorithm that learns what's in a name. *Machine Learning*, 34(1/3).

P. F. Brown, S. A. Della-Pietra, V. J. Della-Pietra, and R. L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2).

Nancy Chinchor. 1997. MUC-7 Named Entity Task Definition. In *Proceedings of the 7th Message Understanding Conference. http://www.muc.saic.com/*.

Gregory Grefenstette. 1999. The WWW as a Resource for Example-Based MT Tasks. In *ASLIB'99 Translating and the Computer 21*.

Andrei Mikheev, Marc Moens, and Calire Grover. 1999. Named Entity Recognition without Gazetteers. In *Proceedings of the EACL*.

Bonnie G. Stalls and Kevin Knight. 1998. Translating Names and Technical Terms in Arabic Text. In *Proceedings of the COLING/ACL Workshop on Computational Approaches to Semitic Languages*.